引入评分偏置的二项矩阵分解推荐算法 *

张笑虹，张奇志，周亚丽

(北京信息科技大学 自动化学院，北京 100192)

摘 要：针对推荐系统中的评分预测问题，在矩阵分解的基础上，实现了一种修正的二项矩阵分解算法。假设用户对物品的评分基于二项分布，由于用户的评分习惯存在差异，物品的受欢迎程度也存在着差异，导致用户—物品评分矩阵存在偏置量。通过引入偏置量对矩阵分解和评分预测进行修正，采用最大后验估计建模，并通过随机梯度下降算法优化模型。实验结果表明，在 MovieLens 100K 数据集上，引入评分偏置的二项矩阵分解算法在推荐精度，离线计算时间等方面均优于传统的二项矩阵分解算法。

关键词：推荐系统；二项矩阵分解；评分偏置

中图分类号：TP301.6 doi: 10.19734/j.issn.1001-3695.2018.10.0807

Binomial matrix factorization with rating drift for recommender systems

Zhang Xiaohong, Zhang Qizhi, Zhou Yali

(*School of Automation*, *Beijing Information Science & Technology University*, *Beijing* 100192, *China*)

Abstract: Based on matrix factorization techniques, in this paper, implemented a modified binomial matrix decomposition algorithm in order to solve the recommender system's rating prediction problem. Suppose the user's rating of the item is based on the binomial distribution. There are differences in the user's rating habits, and there are differences in the popularity of the items, resulting in an offset in the user-item scoring matrix. So, use the maximum a posteriori estimate to design model and the model is optimized by a stochastic gradient descent algorithm. The experimental results show that the modified binomial matrix decomposition algorithm is superior to the traditional binomial matrix decomposition algorithm in terms of recommender accuracy and offline calculation time on the MovieLens 100K datasets.

Key words: recommender system; binomial matrix factorization; rating drift

0 引言

大数据时代的标志之一便是“数据丰富，但信息贫乏”。对于信息消费者，从海量信息中找到自己感兴趣的信息变得十分的困难；对于信息生产者，如何捕捉用户兴趣，洞悉用户兴趣变化，生产出用户感兴趣的信息变得极具挑战[1]。

解决或缓解这种矛盾的核心是通过“漏斗”来减少最终呈献给用户的信息数量。推荐系统就是解决这一矛盾的重要工具，通过对信息进行过滤、排序，呈现给用户有用的信息。推荐系统的本质就是通过一定的方式将用户和物品联系起来，通过发掘用户的需求与兴趣，通过推荐算法从海量数据中挖掘出用户可能感兴趣的项目推荐给用户。目前，推荐系统已经成功应用到了互联网的众多领域中[14]。

传统推荐方法主要包括协同过滤技术[5,6]、基于内容的推荐方法[7,12]和混合推荐方法[8]。文献[2]给出了推荐算法的形式化定义：U 表示用户(*user*)集合，I 表示物品(*item*)集合，定义 f 为效用函数，用来计算项目 i 对用户 u 的推荐度，而推荐算法通过计算推荐度为用户 $u\in \boldsymbol{U}$ 找到其最感兴趣的项目 $i\in \boldsymbol{I}$，如式（1）所示。

$$\forall u\in U, i_u^{'} = \arg\max f(u,i) \tag{1}$$

推荐系统有效性需要解决的一个关键问题是：效用函数 f 通常是定义在 $U\times I$ 的一个子空间上，而推荐算法则必须将 f 外推至整个 $U\times I$ 空间[14]。例如，现在很多应用通常将推荐度定义为用户对物品的评分，但在实际中，由于用户仅仅对一小部分物品进行了评分，这在数据中的表现为稀疏性。在实际应用中，由已知的评分数据通过机器学习算法计算出未知的评分，通过候选和排序过程，最终为用户生成一个 Top-N 列表，这就是外推的过程。在这个外推的过程中，最经典的算法是协同过滤算法：利用用户和物品之间的交互信息为用户产生推荐。其中主要包括基于邻域的方法、隐语义模型(latent factor model，LFM)、基于图的随机游走算法[15]三种方法。这其中最著名、在工业界得到最广泛应用的是基于邻域的方法[5]。

而隐语义模型是最近几年推荐系统领域最为热门的研究话题，其核心是通过隐含特征联系用户兴趣和物品。在推荐系统领域，隐语义模型和矩阵分解模型思想一致，都是通过降维的方式将矩阵补全。最早的矩阵分解模型来源于数学上的奇异值分解(singular value decomposition，SVD)。2006 年 Netflix Prize 开始后， Funk[16]在博客上公布了一个算法(Funk-SVD 算法)，引起了学术界对矩阵分解类方法的强烈关注。此后便出现了一系列改进模型：加入偏置项后的 Biased-SVD 算法、考虑用户历史评分过的物品而得到的 SVD++算法[10]、基于用户评分数据服从正态分布的概率矩阵分解算法、针对离散评分数据集上的二项矩阵分解算法(binomial matrix factorization，BMF) [11]等。

本文主要针对隐语义模型中二项矩阵分解算法存在的某些问题在 MovieLens 100k 电影评分数据集上通过离线实验展开研究。

收稿日期：2018-10-18；**修回日期：**2019-01-10 **基金项目：**国家自然科学基金资助项目（11672044，11172047）

作者简介：张笑虹（1992-），男，湖南沅江人，硕士研究生，主要研究方向为模式识别、推荐系统（zxh715719@163.com）；张奇志（1963-），男，辽宁阜新人，教授，博士，主要研究方向为机器人控制；周亚丽（1968-），女，辽宁沈阳人，教授，博士，主要研究方向为机器人控制、信号处理.

1 传统的矩阵分解算法

矩阵分解算法主要应用于评分预测问题。设存在 I 个商品，N 个用户，$N\times I$ 表示评分矩阵 R，则矩阵 R 中元素 r_{ui} 表示用户 u 对物品 i 的评分。现假设 D 为用户和物品的潜在特征个数，那么 $D\times N$ 维的矩阵 p 表示用户的潜在特征矩阵，p_u 表示用户 u 的潜在特征向量；$D\times I$ 维的矩阵 q 表示物品的潜在特征矩阵，q_i 表示物品 q 的潜在特征向量。由此预测用户 u 对物品 i 的评分为

$$\hat{r}_{ui}=q_i^T p_u \tag{2}$$

所以，评分矩阵 R 便可由两个低秩矩阵 p 和 q 近似的表示为

$$R\approx\hat{R}=q^T p \tag{3}$$

因此，本文认为只要两者误差尽量小，便能用预测值来代替真实值，将评分预测问题转换成最优化问题。

SVD 是最经典的矩阵分解算法之一，基本思想表示为

$$R=U\Sigma V^T \tag{4}$$

其中：U 和 V 分别表示用户物品隐含因子矩阵；Σ代表奇异值矩阵并且为对角矩阵。基于数据中的一小段携带了数据集中的大部分信息，其他信息要么是噪声，要么就是毫不相干的信息的前提下，本文只需选取前 D 个因子即可表示某个用户或物品。在 Netflix Prize 中，SVD 算法及其改进算法表现良好[13]。虽然 SVD 算法简单易于实现，但 SVD 算法要求矩阵是稠密的，即矩阵里的元素要非空，否则就不能进行矩阵分解。对此，传统的 SVD 算法通常用全局平均值对评分矩阵中的缺失值进行简单补全。受制于用户数和物品数，传统矩阵分解算法在实际生产环境中难以使用。

针对 SVD 算法需要填充矩阵，分解降维，特别是矩阵求逆的时间复杂度为 $O(N^3)$的问题， Funk 提出了 Funk-SVD 算法，通过将矩阵分解为低秩的用户、物品矩阵，同时降低计算复杂度，借鉴线性回归思想，最小化观察数据的平方来寻求最优的用户和项目的隐向量表示，如式(5)所示。

$$\min_{q^*,p^*}\sum_{(u,i)\in K}(r_{ui}-q_i^T p_u)^2 \tag{5}$$

并提出 L2 正则化矩阵分解，通过结构风险最小化来降低模型学习过程中的过拟合问题：

$$\min_{q^*,p^*}\sum_{(u,i)\in K}(r_{ui}-q_i^T p_u)^2+\lambda\left(\|q_i\|^2+\|p_u\|^2\right) \tag{6}$$

同时通过随机梯度下降算法来寻求函数的最优解。在获得用户和物品因子矩阵 p 和 q 后，由式(7)计算用户 u 对物品 i 的预测评分。

$$\hat{r}_{u,i}=p_u^T q_i=\sum_{k=1}^{K}p_{uk}q_{ik} \tag{7}$$

算法 1 Funk-SVD 算法

初始化潜在因子数 K、惩罚参数 λ、学习率 η，初始化模型参数 p 和 q(从正态分布中随机抽取出这些值)。

a) 对每个用户—物品评分(u, i)。

(a) 计算评分残差 $e_{u,i}=r_{u,i}-\hat{r}_{u,i}$，其中 $\hat{r}_{u,i}$ 由式(7)计算得到。

(b) 更新用户 u 和物品 i 的因子向量 p_u 和 q_i：

$$p_{u,k}+=\gamma(e_{u,i}\cdot q_{i,k}-\lambda\cdot p_{u,k})$$
$$q_{i,k}+=\gamma(e_{u,i}\cdot p_{u,k}-\lambda\cdot q_{i,k})$$

b) 计算新的均方根误差(root mean square error, RMSE)。如果新的 RMSE 比之前 RMSE 小，则继续前面更新步骤；否则终止算法。

Funk-SVD 算法虽然思想很简单，但是在实际应用中效果比较好，具有容易编程实现、实现复杂度低、预测效果较好等，同时还能保持扩展性等优点。后续许多著名模型都是通过对 Funk-SVD 模型改进而获得[1]。

2 基于修正的二项矩阵分解算法

传统矩阵分解算法利用机器学习的思想有利于克服评分数据的稀疏性问题，通过引入 L2 范数降低模型的过拟合问题，但矩阵分解算法在给定用户和物品的潜在因子后，认为对应的评分随机变量满足正态分布。然而对于绝大部分实际应用协同过滤问题里，评分数据往往是在若干离散的整数间取值[4]。例如 MovieLens 100 k 电影评分数据集和 Netflix Prize 中，评分值 S={1,2,3,4,5}显然采用上述假设不合理。

因此，采用二项分布假设代替矩阵分解算法中的正态分布假设，如式（8）所示。

$$p(R_{u,i}\mid p_u,Q_i)=B(R_{u,i}-1\mid S-1,\beta_{u,i}) \tag{8}$$

其中：$B(k|n, p)$为具有参数 n 和 p 的二项分布函数；S 为界定允许评分范围的定值(MovieLens 100k 和 Netflix Prize，$S=5$)。令

$$\beta_{u,i}=\frac{1}{1+e^{-p_u^T q_i}} \tag{9}$$

β_{ui} 为用户 u 和物品 i 的潜在因子随机向量点积的某函数值。因此，用户 u 对物品 i 的评分 r(r=1,2,......,S)的概率 P 为

$$P(R_{u,i}=r\mid P_u=p_u,Q_i=q_i)=\binom{S-1}{r-1}\beta_{u,i}^{r-1}(1-\beta_{u,i})^{S-r} \tag{10}$$

即假定用户对所有的物品都有打分行为，每次打分至少为 1 分，本文认为用户 u 根据自己的喜好对每个物品打的每 1 分可看成是“喜欢”或“不喜欢”，这样用户对电影的评分满足二项分布，所以 BMF 中 P 和 Q 的 log-后验分布为

$$\begin{aligned}&\log(P,Q\mid R,\theta)=\\&\sum_{(u,i)\in p}\left\{\log\binom{S-1}{r-1}-(s-1)\log(1+e^{-p_u^T q_i})-(S-r_{u,i})p_u^T q_i\right\}-\\&\frac{1}{2\sigma^2}\sum_{u=1}^{U}(p_u-\mu)^T(p_u-\mu)-\frac{1}{2\gamma^2}\sum_{i=1}^{M}(q_i-\theta)^T(q_i-\theta)-\\&\frac{1}{2}[DU\log\sigma^2+DM\log\gamma^2]\end{aligned} \tag{11}$$

令 $\mu=0$ 和 $\theta=0$，选取方差 σ^2 和 γ^2 为合适值[3]，那么最大化式(11)等价于最小化目标函数，如式(12)所示。

$$\begin{aligned}G(P,Q)=&\frac{1}{2}\sum_{(u,i)\in p}\{(s-1)\log(1+e^{-p_u^T q_i})+(S-r_{u,i})p_u^T q_i\}+\\&\frac{\lambda}{2}\left\{\sum_{u=1}^{U}\|P_u\|_2^2+\sum_{i=1}^{M}\|q_i\|_2^2\right\}\end{aligned} \tag{12}$$

获得 P、Q 后，本文使用全新的评分预测公式获得用户 u 对物品 i 的预测评分：

$$\hat{r}_{u,i}=1+\frac{S-1}{1+e^{-p_u^T q_i}} \tag{13}$$

这个公式通过隐类将用户和物品联系在一起。但是在实际情况下，一个评分系统有些固有属性和用户物品无关，而用户也有些属性与物品无关，物品也有些属性和与用户无关。

同时通过对用户对电影的评分数据进行分析，发现 1 682 部电影中的 600 部电影集中了用户 10 万条评分数据中的 83 715 条，即多数用户兴趣集中在这 600 部热门电影中，而其他 1 082 部电影却很少有用户关注甚至没有人关注。也就是说大多数用户只对当中的少数电影有兴趣，而绝大多数的电影只有很少的的人有评分行为。即物品的流行度中存在着长尾分布问题[9]和“哈利波特问题”[1]问题，导致不同物品受欢迎程度也大不一样。针对上述两种情况，本文将实现一种基于修正的二项矩阵分解算法(Biased-BMF)：即在二项矩

阵分解的原始模型中引入用户和物品的评分偏差，同时假设这种偏差信息符合均匀分布或正态分布，则基于评分偏置的用户 u 对物品 i 的评分预测公式为

$$\hat{r}_{u,i}=1+\frac{S-1}{1+e^{-p_u^T q_i+b_u+b_i}} \tag{14}$$

而这种基于评分偏置的二项矩阵分解算法，其对应的最小化目标函数为

$$G(p,Q)=\frac{1}{2}\sum_{(u,i)\in p}[(S-1)\log(1+e^{-p_u^T q_i})+(S-r_{u,i})(p_u^T q_i+b_u+b_i)]+\sum_{(u,i)\in p}\left[\lambda_1(\|p_u\|_2^2+\|q_i\|_2^2)+\lambda_2(b_u+b_i-\bar{r})^2\right] \tag{15}$$

其中：λ_1 和 λ_2 为惩罚参数；$\bar{r}$ 表示训练集中所有记录的评分的全局平均值。在不同的应用中，应用定位和物品的不同，整体评分分布也会显示出一些差异，全局平均值用来表示应用本身对用户评分的影响。b_u 表示用户偏置项，表示用户的评分习惯不受物品因素的影响，即只与用户评分习惯有关。b_i 表示物品偏置项，表示物品接受的评分中和用户没有关系的因素，即只与物品特征有关。同样本文采用随机梯度下降算法来最小化目标函数。使用随机梯度下降算法求解最优解的迭代过程如算法 2。

算法 2 Biased - BMF)

初始化：潜在因子数 D，惩罚参数 λ_1 和 λ_2，学习率 γ，迭代步数 *step*，训练集与测试集分割比 *ratio*，全局评分均值 $\bar{r}$；

初始化：模型参数 P、Q 和评分偏差值 b_u 和 b_i(本文采用从均匀分布中随机抽取这些值的方法获取 P,Q,b_u,b_i 的值)。

a) 对每个用户—物品评分(u，i)。

(a) 计算评分残差 $e_{u,i}=r_{u,i}-\hat{r}_{u,i}$，$\hat{r}_{u,i}$ 由式(14)计算得到。

(b) 更新用户 u，物品 i 的因子向量 p_u 和 q_i：

$$p_{ud}+=\gamma(e_{u,i}\cdot q_{id}-\lambda_1\cdot p_{ud})$$
$$q_{id}+=\gamma(e_{u,i}\cdot p_{ud}-\lambda_1 q_{id})$$

(c) 更新并修正用户 u，物品 i 的偏差：

$$b_u+=\gamma(e_{u,i}-\lambda_2\cdot(b_u+b_i-\bar{r}))$$
$$b_i+=\gamma(e_{u,i}-\lambda_2\cdot(b_u+b_i-\bar{r}))$$

b) 计算新的 RMSE。如果新的 RMSE 小于之前的 RMSE 或者迭代步数小于默认的参数值，则继续更新；否则结束算法。

3 实验结果及评估

评测推荐系统通常的三种方式是离线实验、用户调查、在线实验。由于本文无法提供一个真实的系统环境用于实验，所以采用离线实验。离线实验步骤如下：

a) 数据预处理。对数据集按照一定的方式和格式进行处理，生成一个标准格式的数据集。

b) 对数据集进行分割，按 9：1 的比例生成训练集和测试集。

c) 用训练集训练用户—物品评分模型，在测试集上进行预测。

d) 通过离线指标评价算法预测精度。

本文将采用离线实验的方式在 GroupLens 提供的 MovieLens 100k 电影评分数据集上评估算法。MovieLens 100k 数据集包含 943 个用户对 1 682 部电影的 100 000 条评分数据(1~5)，每个用户至少对 20 部电影进行了评分，稀疏率为 94.11%，每次从数据集中随机抽取 90%的数据作为训练集，10%的数据作为测试集。

本文通过 RMSE 来评估推荐精度，这是目前在离线测试中衡量推荐算法精度的最常见的指标，其值越小，代表算法精度越高，推荐效果也就越好。

设 K 为评分记录条数，$r_{u,i}$ 为真实评分值，$r_{u,i}$ 为评分预测值，则 RMSE 计算方式如式（16）所示。

$$RMSE=\sqrt{\frac{\sum_{(u,i)\in K}(r_{u,i}-\hat{r}_{u,i})^2}{|K|}},\hat{r}_{u,i}=q_i^T p_u \tag{16}$$

同时还通过监测运行时间(*time*)评估算法的运行效率。

通过实验对比 Biased-SVD、SVD ++、BMF 算法及本文提出的 Biased - BMF 算法来判断 Biased-BMF 的性能。该算法主要参数有潜在因子数 D、正则化参数 λ_1 和 λ_2、学习率 γ、迭代步数 *step*、训练集与测试集分割比 *ratio*。 在理论分析的基础上，通过一系列调参，本文在固定学习率 γ=0.02，正则化参数 $\lambda_1=\lambda_2$ = 0.1，分割比 *ratio* = 0.9 的情况下，研究迭代步数 *step* 和潜在因子数 D 在测试集上对推荐算法精度及运行时间的影响。

图 1 和 2 分别代表不同的 *step* 值对算法 RMSE 和 *time* 的影响。从算法的鲁棒性上来看，Biased-BMF 的 RMSE 值波动维持在 0.02 之间，而 BMF 出现了明显的波动。从推荐精度上来看，Biased-BMF 算法的推荐精度明显高于 BMF 及 Biased-SVD，并接近 SVD++算法，特别是当 *step*=300 时，RMSE=0.89，远远优于其他三种算法。从时间或者算法的时间复杂度上来考虑，由于 SVD++算法在模型中引入了如用户历史浏览数据、电影的历史浏览数据等隐式反馈信息，所以本文看到 BMF 和 Biased-BMF 算法所需的时间低于 SVD++和 Biased-SVD 算法。因此认为 Biased-BMF 能够在更短的时间内得到更高的推荐精度，同时推荐精度不会随算法的迭代次数出现大的波动。

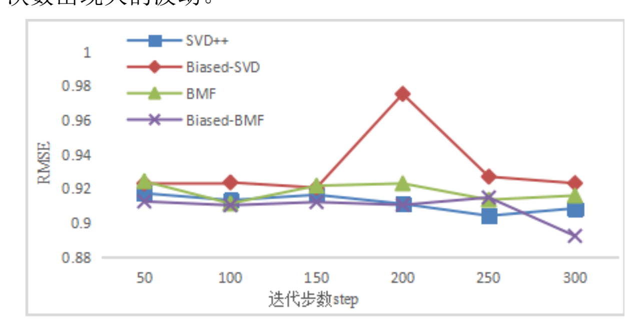


图 1 算法在不同的 *step* 值下的 RMSE 曲线

Fig. 1 RMSE curve of algorithm under different *step* values

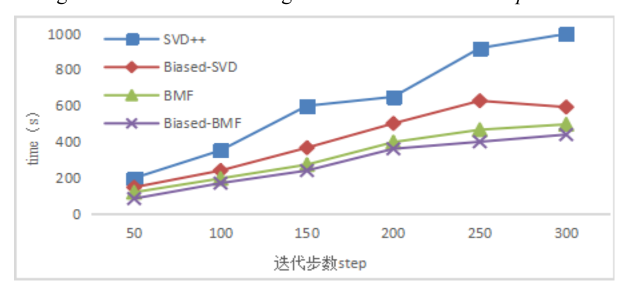


图 2 算法在不同的 *step* 值下的 *time* 曲线

Fig. 2 *time* curve of algorithm under different *step* values

图 3 展示了潜在因子数 D 对预测精度的影响。实验表明，BMF 和 Biased-BMF 推荐精度方面明显优于 BMF 和 Biased-SVD，而 Biased-BMF 整体上优于 SVD++，当 D=300 时，RMSE(Biased-BMF)=0.89，RMSE(BMF)= 0.91。

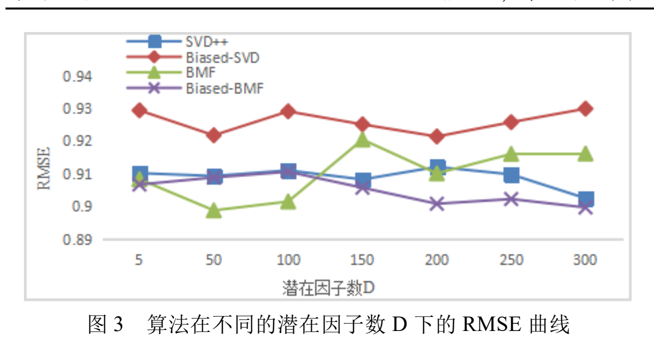


图 3 算法在不同的潜在因子数 D 下的 RMSE 曲线

Fig. 3 RMSE curve of algorithm under different potential factor numbers D

由于 SVD++算法不仅考虑到邻域模型，同时还对未知的评分数据做基准评分预测，从而准确率较高。而 Biased-BMF 则通过深入考虑用户和物品评分偏置量，挖掘数据间的潜在语义信息。因此其预测精度接近于 SVD++算法，但在算法的时间复杂度上优于 SVD++算法。因此，当本文确定潜在因子数 D 时，Biased-BMF 更能精确的表示物品或用户的潜在特征矩阵，提高推荐精度。

4 结束语

本文对基于矩阵分解的协同过滤推荐算法的基本思想和相关工作进行回顾和总结之后，在 BMF 算法的基础上，为了研究用户和物品评分偏置对推荐精度的影响，同时考虑到评分数据为离散值得特点，实现了一种考虑偏置信息的 Biased-BMF 算法。在 MovieLens 100K 数据集上的实验结果表明，Biased-BMF 算法在推荐精度和算法鲁棒性等方面均优于原始的 BMF 算法及其他几种经典的矩阵分解算法。尽管本文对矩阵分解推荐算法中的评分偏置相关问题进行了较深入的研究，但还有很多工作要做，比如融合用户兴趣变化的 Biased-BMF 动态模型；引入深度学习思想更进一步挖掘用户和物品的隐藏特征。

参考文献：

[1] 项亮. 推荐系统实践 [M]. 北京：人民邮电出版社，2012: 24-26. (Xiang Liang. Recommender systems in action [M]. Posts and Telecom Press, 2012: 24-26.)

[2] Tuzhilin A, Adomavicius G. Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions[J]. IEEE Trans on Knowledge and Data Engineering, 2005, 17 (6): 734-749.

[3] Salakhutdinov R, Mnih A. Probabilistic matrix factorization [C]// Proc of International Conference on Neural Information Processing Systems. [S.l.]: Curran Associates Inc, 2007: 1257-1264.

[4] 刘凤林，胡雪蕾. 二项矩阵分解在离散评分推荐算法中的改进 [J]. 计算机应用与软件，2016，33 (1): 81-84. (Liu Fenglin, Hu Xuelei. Improvement of binomial matrix decomposition in discrete scoring recommender algorithm [J]. Computer Applications and Software. 2016, 33 (1): 81-84.)

[5] Smith B, Linden G. Two decades of recommender systems at Amazon. com [M]. [S.l.]:IEEE Educational Activities Department, 2017, 21 (3): 12-18

[6] He X, Liao L, Zhang H, *et al.* Neural collaborative filtering [C]//Proc of International World Wide Web Conference Committee .2017: 173-182.

[7] Mooney R J, Roy L. Content-based book recommending using learning for text categorization [C]// Proc of the 5th ACM Conference on Digital Libraries. New York: ACM Press, 2000: 195-204.

[8] Balabanović M, Shoham Y. Fab: content-based, collaborative recommendation [J]. Communication of the ACM, 1997, 40 (3): 66-72.

[9] 张莉，余磊. 推荐系统中谁可以协同新用户？[J]. 计算机科学，2015，42 (b11): 80-82. (Zhang Li, Yu Lei. Who can cooperate with new users in the recommender system? [J]. Computer Science. 2015, 42 (b11): 80-82.)

[10] Koren Y. Factor in the neighbors: Scalable and accurate collaborative filtering [J]. ACM Trans on Knowledge Discovery from Data, 2010, 4 (1): 1-24.

[11] 吴金龙. Netflix Prize 中的协同过滤算法 [D]. 北京：北京大学，2010. (Wu Jinlong. Collaborative filtering algorithm in the Netflix Prize [D]. Beijing: Peking University, 2010.)

[12] Wu C Y, Ahmed A, Beutel A, *et al.* Recurrent recommender networks [C]// Proc of the 10th ACM International Conference on Web Search and Data Mining. New York: ACM Press, 2017: 495-503.

[13] 李悦，李鹏,等译. 机器学习实战 [M]. 北京：人民邮电出版社，2013: 253-257. (Li Yue, Li Peng, *et al.* Machine learning in action [M]. Beijing: Posts and Telecom Press, 2013: 253-257.)

[14] 黄立威，江碧涛，吕守业，等. 基于深度学习的推荐系统研究综述 [J]. 计算机学报，2018，41(7): 1619-1647. (Huang Liwei, Jiang Bitao, Lyu Shouye, *et al.* A review of research on recommender systems based on deep learning [M]. Chinese Journal of Computers, 2018, 41(7): 1619-1647.)

[15] 赵海燕，张健，曹健. 基于主题分组与随机游走的 App 推荐算法 [J]. 计算机应用研究，2018，35(8): 2277-2280 . (Zhao Haiyan, Zhang Jian, Cao Jian. App recommendation algorithm based on topic grouping and random walk [J]. Application Research of Computers. 2018 (8) , 35(8): 2277-2280.)

[16] Funk S. http://sifter. org/~simon/journal/20061211. html[EB/OL].